

Illinois Institute of Technology

Department of Computer Science

Midterm 2  
CS 440— Programming Languages  
Spring 2006  
November 6, 2006 11:25–12:40

This is a **closed book** and **closed notes** exam.  
You are **not** allowed to use calculators or computers during this exam.  
Do **ALL** problems in this booklet. Read each question very carefully.  
You may detach pages, but **you must return all pages of this exam.**

Name

Email ID

@iit.edu

Do **not** place your social security number anywhere on this exam.

Problem	Points	Score
1	16	
2	6	
3	6	
4	4	
5	6	
6	6	
7	6	
8	6	
9	6	
10	6	
11	6	
12	4	
13	6	
14	6	
15	0	
Total	90	
Percent	100	

## Gramamrs

### Question 1) (16 points)

Consider the following grammar.

$$S \rightarrow x E y$$

$$E \rightarrow E y$$

$$E \rightarrow p$$

$$\text{Follow}(S) = \{\$ \}$$

$$\text{Follow}(E) = \{y\}$$

Generate the Characteristic Finite State Machine, and fill in the LR table. There are 5 states, from 0–4. The first state has been done for you.

$I_0$	$S \rightarrow \cdot x E y$	State	x	y	p	\$	S	E
		0						
		1						
		2						
		3						
		4						

**Question 2)** (6 points) Is the grammar on the previous page ambiguous? Give a proof with your answer.

**Question 3)** (6 points) Is the grammar on the previous page LL? Why or why not?

**Question 4)** (4 points) When faced with a shift-reduce conflict, most parsers will shift. Why do we make that choice?

## Parameters

**Question 5)** (6 points) Consider the following code:

```
let i = 10;;  
let f a =  
  a := 50;  
  a := a + i;  
  i;;  
print f(i);;  
print i;;
```

What will be printed to the screen for the following parameter passing styles?

**Call By Value**

**Call By Reference**

**Call By (Copy-)Result**

**Question 6)** (6 points)

Consider the following code:

```
let f a b =  
  a + a + a;;
```

```
f (10 * 20) (f (20 * 30) (30 / 40));;
```

1. How many times is `*` called in the call by name parameter passing style?
2. How many times is `*` called in the call by need parameter passing style?

**Question 7)** (6 points) What is an advantage of call by need or call by name over call by value?

**Question 8)** (6 points) In a call stack, what is the difference between the static link and the dynamic link? Give an example of a language that would need both of these, **and** another example of a language that would **not** need both of these.

**Question 9)** (6 points) What is the difference between static binding and dynamic binding?

**Question 10)** (6 points) Give an example of a language that has both static binding and dynamic binding of a variable's value. How does the programmer select which kind to use?

## Unification

**Question 11)** (6 points) Solve the following unification problem. Show your work to allow for partial credit if something goes wrong. The Greek letters are the variables.

$$\{g(\alpha, y) = g(x, \beta), \quad h(\gamma, z) = h(f(\alpha), z)\}$$

**Question 12)** (4 points) Give an example of a language feature (from any language) which is implemented using unification.

## Miscellaneous

**Question 13)** (6 points)

Why is the lambda-calculus useful?

**Question 14)** (6 points) The Fortran language has static binding of variable location, this is an unusual choice. Why was it decided to do things this way? Also, what is the cost (i.e., what language feature(s) are unavailable as a result)?

**Question 15)** (0 points) **Extra Credit!**

The following code comes from the interpreter shown in class.

```
| AppExp(e1,e2) ->
  let e1' = eval e1 in
  (match e1' with
  | FunExp(x,m) -> eval (subst m e2 x)
  | _ -> AppExp(e1',e2))
```

What is the parameter passing style implemented by this code?

This code can be modified very simply to make it use call-by-value parameter passing style. Show the modification.